In the Claims:

Please cancel claims 1-7 and add new claims 8-16, all as shown below.

1 – 7. (Cancelled)

8. (New): A method, comprising:

communicating with a resource manager from an application using an Application Programming Interface (API), wherein the API utilizes a logical connection to the resource manager;

controlling transaction demarcation using the Java™ Transaction API (JTA);

communicating with the resource manager from a first transaction manager during two phase commit processing using an XAResource interface;

enlisting a resource, wherein the first transaction manager associates a unique transaction identifier with work that is performed on a resource by invoking XAResource.start() on the resource and subsequent application updates to the resource are associated with a global transaction;

delisting a resource, wherein the first transaction manager invokes XAResource.end() on the resource and future application updates on the resource over the logical connection are disassociated from the global transaction; and

blocking a second transaction manager from calling XAResource.start() on the resource until the first transaction manager has called XAResource.end() on the resource.

9. (New): The method of claim 8, wherein the application communicates to the resource manager using JDBC™ or JMS.

10. (New): The method of claim 8, wherein the first transaction manager and the second transaction manager are multiple threads of the same transaction manager.

11. (New): The method of claim 8, wherein each XAResource instance is wrapped in an object that a transaction manager will use to synchronize concurrent enlistment requests.

12. (New): The method of claim 11, wherein the transaction manager maintains a collection of wrapped XAResource objects which is consulted on each resource enlistment.

13. (New): The method of claim 12, wherein each request to enlist the resource will first check to see if there is a lock being held on the resource by another thread of control.

14. (New): The method of claim 13, wherein if a resource is not yet locked, a lock is granted to an accessor and held until the owner of the transaction ID delists the resource.

15. (New): The method of claim 14, wherein waiting threads are signaled when a lock is freed and one of the waiting threads will be granted the lock and allowed to proceed with its enlistment.

16. (New): The method of claim 15, wherein the collection of wrapped XAResource objects is periodically garbage collected to clear stale and unused entries.